DEVELOPMENT OF A MULTI-AGENT PLATFORM FOR SUPPLY CHAIN-WIDE ORDER FULFILMENT

Roberto Domínguez Cañizares and Jose M. Framiñán
Industrial Management, School of Engineering, University of Seville,
Camino de los Descubrimientos, s/n, Isla de la Cartuja, 41092, Sevilla, Spain

Keywords: Supply Chain Management, Order Fulfilment Process, Multi-Agent System, Simulation.

Abstract: In this paper we describe an agent-based framework for modelling and simulating different processes taking place in supply networks, resulting in a supply chain simulator called SCOPE (*Sistemas COoperativos para la Programación y Ejecución de pedidos*). The framework is composed of reusable elements (agents and objects) allowing easy modelling of real-scale supply chains, with different companies and products. Each company in the model can use different policies and parameters for the different business functions. The framework is implemented using Swarm. Furthermore, its generic and modular structure allows to easily adding new and more complex functions for the agents. The final aim of SCOPE is to serve as a testbed to implement and analyse the effects of different management decisions related to order fulfilment over real-scale supply chains. SCOPE has been validated using different supply chains described in the literature.

1 INTRODUCTION

In today's dynamic and complex manufacturing environment, an important enabler in gaining competitiveness is the ability of a company to respond quickly and effectively to satisfy customers (Framinan, 2009) via an effective Order Fulfilment Process (OFP), which starts with receiving orders from the customers and ends with having the finished goods delivered (Lin et al., 1998).

OFP is a complex process because it is usually composed of several activities, executed by different functional entities heavily interdependent among the tasks, resources, and agents involved in the process (Lin and Shaw, 1998). Furthermore, manufacturing practice is shifting towards the outsourcing paradigm, so OFP activities may take place across different companies, thus hindering the centralized planning and scheduling (Lin and Lin, 2006). Therefore, OFP is likely to be executed over a Supply Chain Network (SCN).

Available-To-Promise (ATP) systems refer to a variety of methods and tools to enhance the responsiveness of order promising and the reliability of order fulfillment (Framinan, 2009). It is difficult to predict the effects of new management policies and strategies to improve OFP on a real SCN, but at the same time this may be of great benefit for companies. One of the most popular approaches is using analytic models, like linear programming, integer/mixed integer programming to analyze the performance of a SCN. However, a SCN involves stochasticity and uncertainty features not suited for these analytical models. In addition, analytical models may not be solved due to their complexity and nonlinearity. Simulation, especially the multi-agent-based distributed simulation, turns out to be one of the most effective tools to model and analyze SCNs (Long et al., 2011).

Because of its ease for modeling and comprehensive description of complex systems (particularly its capability of handling their dynamics and stochastic behaviour), simulation has been widely used in SCN management. There is a great interest in modeling SCNs as Agent-Based Systems (ABS) because there is a natural correspondence between SCN participants and agents in a simulation model. In addition, SCNs tend to be decentralized systems with the participants acting independently, according to their own interests and policies (Long et al., 2011). Thus, the use of an agent-based approach is suitable to model and simulate SCNs (Chatfield et al., 2007).

The rest of this paper is summarized as follows: Section 2 describes the framework, Section 3 describes the implementation of the framework,

Domínguez Cañizares R. and M. Framiñán J..
DEVELOPMENT OF A MULTI-AGENT PLATFORM FOR SUPPLY CHAIN-WIDE ORDER FULFILMENT.
DOI: 10.5220/0003747802030208
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 203-208
ISBN: 978-989-8425-96-6
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

Section 4 is validation of the simulation platform obtained (SCOPE) and Section 5 is the conclusion.

2 FRAMEWORK

Real SCNs have multiple layers of abstraction (Lin et al., 2002), as they can be studied in different levels of details. Thus, we model two different layers: an Enterprise Layer containing all enterprises in the SCN, and a Functional Layer, including the main functions/departments of the enterprises. This structure will allow studying inter-enterprises relationships and intra-enterprises relationships.

The Enterprise Layer is modelled by one generic and reusable agent (Enterprise Agent) composed of several functional agents modelling physical and planning tasks. By doing this, every department in the enterprise is encapsulated in one agent, with its characteristics of independency and autonomy, and being able to take its own decisions.

A central feature of ABS is the bottom-up methodology to construct a model. In this methodology the user assumes that he/she cannot understand the whole phenomenon of interest but can observe, at a micro level, specific activities and processes, and tries to understand their behaviour and their objectives. These agents interact and communicate with other agents and they join to form a coherent whole on a macro level (Nilsson and Darley, 2006). This whole is the emergent behaviour, which cannot be predicted in advance.

In accordance with the bottom-up methodology, SCN and the Enterprise Agent are not explicitly modelled. Instead, the Enterprise Agent behaviour emerges from its components' behaviours (i.e., functional agents), which are easier to understand and model. Similarly, the global SCN behaviour emerges from that of its components enterprises.

2.1 Enterprise Agent

The Enterprise Agent is able to model any kind of company in the SCN. The composition of their functional agents determines its behaviour. To simplify the Enterprise Agent configuration we identify four roles of the companies in the SCN. Companies with similar functions (in terms of functional agents) belong to the same category. These four role categories and their main characteristics are summarized in the Table 1.

To model a new company, it is enough to select one of the roles from Table 1 for the Enterprise Agent and automatically the required functional agents are assigned to it. The basic agents always belong to the Enterprise Agent for the selected role, while the optional agents are selected depending on the enterprise characteristics.

2.2 Functional Agents

A good design of the functional agents is crucial because they must capture the internal dynamic of a real enterprise and should model the key aspects of enterprise management. We base our choice in the level 1 of the SCOR model (SCC, 2006) and the literature revised.

Level 1 of the SCOR model summarises business operations in five main activities: Plan, Source, Make, Deliver and Return. Activities are divided in two groups: Physical activities (Source, Make, Deliver and Return) to manage the physical resources of the enterprise, and planning activities (Plan) to make decisions. Each one of the physical activities (with the exception of Return) is modelled by one agent, so there are three physical agents:

Table 1: Roles of the Enterprise Agent.

Roles in Framework	Description	Examples types	Basic Agents	Optional Agents
Manufacturer	The enterprise is in the SCN and has manufacturing or assembly capacity	Manufacturer, Assembler	Demand Fulfillment, MRP, Scheduling, Source, Make, Deliver	Demand Forecast, Master Planning, Production Planning
Intermediate	The enterprise is in the SCN and has not manufacturing or assembly capacity	Distributor, Wholesaler, Retailer	Demand Fulfillment, MRP, Deliver	Demand Forecast
External Provider	Any enterprise who provide something to the SCN, but it's out of the SCN	Provider	Demand Fulfillment, Deliver	No
External Customer	Any enterprise who demand something to the SCN, but it's out of the SCN	Customer	MRP, Source	No

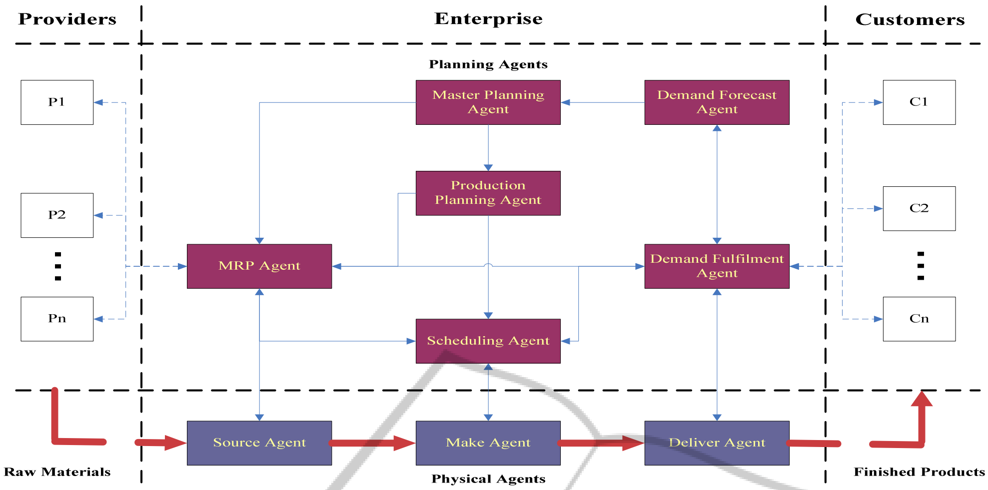

Figure 1: Multi-Agent Framework: general configuration of the Enterprise Agent.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Source Agent, Make Agent and Deliver Agent. The Return activity is implicitly implemented by allowing the Deliver Agent to return products or receive returned products. According to the Supply Chain Planning Matrix in Stadtler (2005), the Plan activity has been divided into six planning functions, being each of these functions carried out by a different agent: Demand Fulfilment Agent, Demand Forecast Agent, Master Planning Agent, Production Planning Agent, MRP (Material Resource Planning) Agent and Scheduling Agent.

The enterprise is then modelled as a mix of planning agents and physical agents. Planning agents store management policies and take the main decisions. Physical agents control the physical resources of the enterprise and share information with planning agents. The overall configuration of the Enterprise Agent, with all its functional agents is shown in Figure 1. A brief description of the agents is presented below:

- Source Agent, to handle the arrival and storage of raw materials, and its delivery to the manufacturing process when needed.
- Make Agent, to monitor the manufacturing process by controlling machines and the flow of jobs in the shop floor.
- Deliver Agent, to handle the arrival and storage of finished products, and the delivering of orders to customers.
- Demand Fulfillment Agent, in charge of demand management and inventory control. It checks incoming RFQs (Request for Quotations) from customers and quotes the due dates. If the order is accepted, it tries to fulfill them from inventory, if available. If inventory is not enough, it sends a production order (a Job) to the Scheduling Agent if the enterprise is a manufacturer. Otherwise it sends a purchase order to the MRP Agent.
- Demand Forecast Agent: This agent requests the enterprise demand historical data to the Demand Fulfillment Agent in every forecast period. Then it forecasts the demand for each product in the next periods using a forecasting rule (like Simple Moving Average, etc.).
- Master Planning Agent: In every planning period, this agent uses forecast information from the Demand Forecast Agent and generates an aggregate Master Plan for the products concepts defined by the company by solving a linear programming model (that includes capacity and inventory restrictions, as well as production and inventory holding costs), obtaining the production needs per product concept for each period.
- Production Planning Agent: It receives the Master Plan and takes only the production needs for the first planning period. Then it generates a detailed Production Plan solving one model for each product concept and obtaining the production needs for the final products to accomplish the Master Plan.
- MRP Agent: It creates a detailed material plan to fulfil the Master Plan. If there is no Master Plan, it takes control of the raw material inventory levels by using some inventory policy. This agent is the one in charge of the purchasing functionality in the enterprise.
- Scheduling Agent: This agent schedules production orders (Jobs) coming from the Production Planning/Demand Fulfillment

Agent by using first some priority rule to create an initial solution, and then a heuristic to improve the initial solution according to certain objective. It also calculates starting and ending times for each job (so it can help the Demand Fulfillment Agent for due date calculation), generating a detailed schedule. According to this schedule, at the starting time of each job it sends the job information to the Make Agent to start its production.

3 IMPLEMENTATION

SCOPE has been implemented in Swarm (Java version) and NetBeans IDE 6.7 as implementation framework. Swarm is a multi-agent software platform for the simulation of complex adaptive systems (Minar et al., 1996). It provides object oriented libraries of reusable components for building models and analyzing, displaying, and controlling experiments on those models. Swarm is suitable for modeling SCN (Lin et al., 1998). In Lin et al., (2002) we can see a comparison between SCN's features and Swarm. Furthermore, models are coded in Java, so low-level functions can be added.

We have considered other platforms for the implementation of our framework, such as NetLogo or Repast. NetLogo stands out for its ease of use, but it is basically designed for mobile agents acting concurrently on a grid space with behaviour dominated by local interactions over short times, and that are not extremely complex (Railsback et al., 2006). Repast was initially conceived for implementing Swarm in Java. However, it does not implement swarms, which are very helpful for organizing models. Furthermore the schedule executes top-level actions in randomized order (which is not desirable), while Swarm allows a precise control of the sequence of actions.

In our model we have four different types of objects, nine agents and three swarms. **Objects** are information containers created by the agents, who use them and send them to other agents. A typical object is represented by a java class, where all variables and methods are defined. The objects available are: RFQ, Job, Product and Machine.

Agents are the basic elements in the simulation model. They represent the main functions in the enterprise (functional agents in the framework). Their behavior is modeled by writing methods. Information is passed by arguments, which makes easy to add new capabilities to the agents, by simply adding new methods, or overwriting existing ones. A typical agent in our model is represented by a java class that extends the Agent class in Swarm. A pseudo code example is shown below:

```
   Public class AgentName extends
SwarmObjectImpl {
   //Internal Variables
   Private Type name1;
   Private Type name2;
   ...
   //Constructor
   Public AgentName (Type name1, Type
name2,...) {
   this.name1=name1;
   this.name2=name2;
   ...
   }
   /*Methods: define behaviours and
abilities of the agent*/
   Public Type behaviour1 (Type name3) {...}
   Public void behaviour2 () {...}
   ...
   }
```

Swarms are agent's containers with schedules of actions. The three swarms in the model are the following:

- Enterprise: this is the main swarm, and it serves to model an entire enterprise. It is formed by a combination of the nine types of agents described before and contains the schedules for all these agents, controlling their actions. The behaviour of the enterprise swarm is defined by the emergent phenomena of the agents inside themselves.
- Model: it models the SCN environment, and it contains all the enterprises.
- Observer: it contains the model swarm and special methods to present all relevant information from the agents in the model swarm, for a post-simulation analysis.

For solving the planning models included in the Master Planning and Production Planning agents, SCOPE can be connected with Gurobi solver through a special library for Java. Gurobi is a commercial software package for solving large-scale mixed-integer linear optimization problems.

4 VALIDATION

In order to validate SCOPE we have looked into the literature for SCNs modeled and simulated by other authors and have compared their results with those provided by our platform. Particularly, in Chatfield et al. (2004) their Simulator for Integrated Supply Chain Operations (SISCO) is validated by comparing their results with those by Chen et al. (2000) and Dejonckheere et al. (2003). Therefore,

SCIENCE AND TECHNOLOGY PUBLICATIONS

we validate SCOPE using the same two scenarios as in Chatfield et al. (2004), and simulate a third scenario reproducing some of the experiments conducted in Chatfield et al. (2004) after the validation of SISCO.

The SCN's structure is the same for the first two scenarios: Only one product, and SCN structured by a serial formation of customer, retailer, wholesaler, distributor, and factory levels. The lower node places orders with the next upper node and this node fills these orders. The customer does not fill orders and the factory places orders with an outside supplier. There are other features that remain identical for both scenarios, like the use of a fixed lead time, or the forecast rule and inventory policy. For more details see Chatfield et al. (2004).

4.1 Scenario 1: Chen et al. (2000)

Chen et al. calculated a statistical lower bound for the variance amplification in the SCN described above, and obtained the result shown in equation (1).

$$\frac{Var(q^k)}{Var(D)} \geq \prod_{i=1}^{k}\left(1+\frac{2L_i}{p}+\frac{2L_i^2}{p^2}\right), \forall k \quad (1)$$

We have obtained very similar results to those offered by SISCO, although we found a greater amplification at upper SCN stages than in Chen et al. (2000). Chatfield et al. justify these results arguing that the bounds provided by Chen et al. do not account for interactions and interdependencies present in a multi-stage system. To test this, they perform a "sequential pairs execution" simulation, in which they broke the supply chain into four two-node sub-chains (customer-retailer, retailer-wholesaler, wholesaler - distributor, distributor - factory). Then, they simulate each sub-chain using the ordering mean and standard deviation obtained from the simulation of the previous sub-chain. We reproduced this new scenario, obtaining the results in Table 2, which are extremely close to those predicted by Chen et al., and SISCO.

Table 2: Amplification Ratio for Chen et al. (2000), "sequential pairs execution" experiment. Parameters: demand rate =□$N(50, 20^2)$; protection time = L+R = 4+1 = 5; MA(15) forecasting; simulation time = 5200 time periods (200 for warm-up).

Enterprise	Chen et al.	SISCO	SCOPE
Retailer	1.89	1.90	1.90
Wholesaler	3.57	3.59	3.53
Distributor	6.74	6.70	6.66
Factory	12.73	12.84	12.58

4.2 Scenario 2: Dejonckheere et al. (2003)

These authors used a Control Engineering methodology to study the variance amplification in the SCN described before, and obtained the result shown in equation (2). The results for this scenario are summarized in Table 3, where it is possible to notice that SCOPE performs very similar to SISCO and to Dejonckheere et al. (2003).

$$TF_n = \left[\frac{-2 - Tp + 2z^{Tm} + Tmz^{Tm} + Tpz^{Tm}}{Tmz^{Tm}}\right]^n \quad (2)$$

Table 3: Amplification Ratio for Dejonckheere et al. (2003). Parameters: demand rate = $N(100, 10^2)$; protection time = L+R = 3+2 = 5; MA(19) forecasting; simulation time = 5200 time periods (200 for warm-up).

Enterprise	Dejonckheere et al.	SISCO	SCOPE
Retailer	1.67	1.67	1.71
Wholesaler	2.99	2.99	3.10
Distributor	5.72	5.72	5.96
Factory	11.43	11.43	11.93

4.3 Scenario 3: Chatfield et al. (2004)

After the validation of SISCO these authors studied the impact of information sharing and different degrees of information quality on the Bullwhip Effect when the lead time between companies is stochastic. The SCN used is similar to the one used in the previous scenarios, with only some differences like the mentioned stochastic lead time or the inventory policy used (see Chatfield et al. 2004 for more details). We have used SCOPE to reproduce two of these experiments. The first experiment analyzes the influence of different information quality levels on the Bullwhip Effect for a lead time c.v. = 0.50. The results obtained by SCOPE (Figure 2) are identical to the results obtained by SISCO in Chatfield et al. (2004). A second experiment analyzes the impact of the variance of the lead time on the standard deviation of orders for a given information quality level. Again the results obtained by SCOPE were identical to those by SISCO. Therefore we can conclude that the validation of SCOPE is successful.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

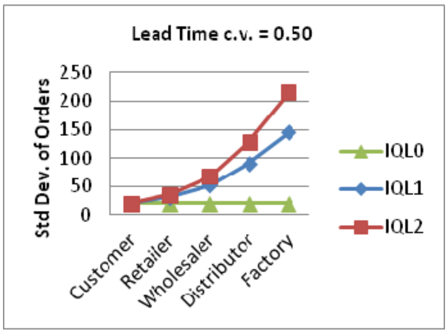

Figure 2: SCOPE results for Scenario 3.

5 CONCLUSIONS

We have developed SCOPE, a framework with a modular design for SCN simulation and analysis. This tool may help SCN managers and researchers to better understand how a given SCN configuration performs in the presence of external and/or internal disturbances. SCOPE is completely open for improvement, which can be done in several ways:

- New functions, i.e. more policies, planning models, heuristics methods, priority rules, forecast methods, etc.
- New behaviours in order to have agents more proactive and with negotiation abilities.
- Including a User Interface.

Future work includes the following lines:

- Studying the Bullwhip Effect and optimizing inventory policies when there are multiple Providers, each one providing different raw materials with different stochastic lead times.
- Testing up-to-date heuristic rules for scheduling. Simulation of full-scale SCNs and analysis of the impact that these have on lead times and customer satisfaction when they are implemented at different levels of the SCN.
- Studying different policies for purchase selection, giving to the agents the ability of selecting the best offer in each purchase and analyzing their individual and global benefits.

REFERENCES

Chatfield, D., Kim, J., Harrison, T., Hayya, J., 2004. *The Bullwhip Effect—Impact of Stochastic Lead Time, Information Quality, and Information Sharing: A Simulation Study.* Production and Operations Management, Vol. 13(4), pp. 340-353.

Chatfield, D., Hayya, J., Harrison, T., 2007. *A multi-formalism architecture for agent-based, order-centric supply chain simulation.* Simulation Modelling Practice and Theory, Vol. 15(2), pp. 153-174.

Chen, F., Drezner, Z., Ryan, J., Simchi-Levi, D., 2000. *Quantifying the bullwhip effect in a simple supply chain: the impact of forecasting, lead times, and information*. Management Science, Vol. 46(3), pp. 436-443.

Dejonckheere, J., Disney, S., Lambrecht, M., Towill, D., 2003. *The impact of information enrichment on the Bullwhip effect in supply chains: A control engineering perspective*. European Journal of Operational Research, Vol. 153, pp. 727-750.

Framinan, J. M, 2009. *Managing resources for order promising in Available-To-Promise (ATP) systems: A simulation study.* International Conference on Industrial Engineering and Systems Management.

Lin, F.-R., Tan, G. W., Shaw, M. J., 1998. *Modeling Supply-Chain Networks by a Multi-Agent System.* Proceedings of the Hawaii International Conference on System Sciences, Vol. 5, pp. 105-114.

Lin, F.-R., Shaw, M., 1998. *Reengineering the Order Fulfillment Process in Supply Chain Networks.* International Journal of Flexible Manufacturing Systems, Vol. 10 (3), pp. 197-229.

Lin, F.-R., Huang, S.-H., Lin, S.-C, 2002. *Effects of Information Sharing on Supply Chain Performance in Electronic Commerce.* IEEE Transactions on Engineering Management, Vol. 49 (3), pp. 258-268.

Lin, F.-r., Lin, Y., 2006. *Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders.* Electronic Commerce Research and Applications, Vol. 5(4), pp. 313-322.

Long, Q., Lin, J., Sun, Z., 2011. *Modeling and distributed simulation of supply chain with a multi-agent platform.* International Journal of Advanced Manufacturing Technology, pp. 1-12.

Minar, N., Burkhart, R., Langton, C., Askenazi, M., 1996. *The Swarm simulation system: A toolkit for building multi-agent simulations.* Working Paper 96-06-042, Santa Fe Institute, Santa Fe.

Nilsson, F., Darley, V., 2006. *On complex adaptive systems and agent-based modelling for improving decision-making in manufacturing and logistics settings: Experiences from a packaging company.* International Journal of Operations and Production Management, Vol. 26 (12), pp. 1351-1373.

Railsback, S., Lytinen, S., Jackson, S., 2006. *Agent-based Simulation Platforms: Review and Development Recommendations.* Simulation, Vol. 82(9), pp. 609-623.

SCC, 2006. *SCOR v8.0*. Supply Chain Council, Inc, Washington.

Stadtler, H., 2005. *Supply chain management and advanced planning - Basics, overview and challenges*. European Journal of Operational Research, Vol. 163 (3), pp. 575-588.